Tutorial

Tutorial REALbasic

Documentación por Javier Menéndez.

Concepto por Geoff Perlman.

© 1999-2009 por REAL Software, Inc. Todos los derechos reservados.

Impreso en EE.UU.

Dirección Postal	REAL Software Inc, PO Box 162181 Austin, TX 78716
Sede Web	http://www.realsoftware.com
Sede ftp	ftp://ftp.realsoftware.com
Soporte	Enviar a través de www.realsoftware.com
Bugs/Solicitud de características	Enviar a través de www.realsoftware.com
Ventas	sales@realsoftware.com
Teléfono	512-328-REAL (7325)
Fax	512-328-7372

Versión 2009 Release 3, julio de 2009.

Tutorial REALbasic

¡Bienvendido al Turorial de REALbasic!

Este *Tutorial* está dirigido a las personas que sean noveles en el mundo de la programación y también noveles en el uso de REALbasic. Te proporcionará una introducción al entorno de desarrollo de REALbasic, te guiará a través del desarrollo de una aplicación, y te mostrará el tipo de otras aplicaciones que pueden crearse mediante REALbasic.

El *Tutorial* asume que ya has completado la guía de *Inicio Rápido*. Completar este *Tutorial* debería llevarte en torno a una hora.

Nota. Si tienes experiencia con otras versiones del lenguaje BASIC o bien otros lenguajes de programación, entonces querrás consultar la Guía del Usuario.

Convenciones de presentación

El *Tutorial* utiliza capturas de pantalla obtenidas de las versiones de REALbasic para Windows, Macintosh y Linux. El diseño de la interfaz y el conjunto de características son idénticos en todas las plataformas, de modo que las diferencias entre cada una de las plataformas son las correspondientes a la interfaz Aero de Vista, la interfaz Leopard de Mac OS X y la distribución Ubuntu de Linux.

Se utiliza el estilo *cursiva* para destacar la primera vez que se utiliza un nuevo término y para señalar el uso de conceptos importantes. Adicionalmente, también se utiliza el estilo cursiva sobre los títulos de los libros, como por ejemplo *Guía de Usuario de REALbasic*.

Cuando se te indique que has de seleccionar un elemento desde los menús de REALbasic, verás algo como "selecciona Archivo > Nuevo Proyecto". Esto es equivalente a "selecciona Nuevo Proyecto en el menú Archivo".

Los elementos contenidos entre paréntesis son los *atajos de teclado* y están formados por una secuencia de teclas que deben pulsarse en el orden mostrado. En Windows y Linux, la tecla Ctrl es la modificadora; en Macintosh, la tecla Comando es la tecla modificadora. Por ejemplo, cuando veas el atajo "Ctrl - O" o "Comando - O", significa que debes pulsar la tecla Control en un ordenador Windows o Linux y pulsar a continuación la tecla "O", o bien pulsar la tecla Comando en Macintosh y pulsar a continuación la tecla "O". Debes soltar la tecla modificadora una vez hayas pulsado la tecla de atajo.

El estilo **Negrita** se utiliza para indicar el texto que debes escribir durante el uso de REALbasic.

Algunos pasos te pedirán que introduzcas líneas de código en el Editor de Código de REALbasic. Estas aparecerán en Helvetica (una fuente sin serifa) dentro de una caja sombreada:

```
ShowURL SelectedURL Text
```

Cuando introduzcas código, por favor, ten en cuenta las siguientes reglas:

- Escribe cada línea en una línea independiente dentro del Editor de Código. No intentes encajar dos o más líneas impresas sobre la misma línea, o bien dividir una línea muy larga en dos o más líneas.
- No añadas espacios adicionales donde no existan sobre el código impreso.

Cada vez que ejecutes tu aplicación, REALbasic comprobará en primer lugar tu código en busca de errores de sintaxis y ortográficos (sobre el código). Si dicha comprobación encuentra algún error, REALbasic dirigirá directamente tu atención sobre la línea de

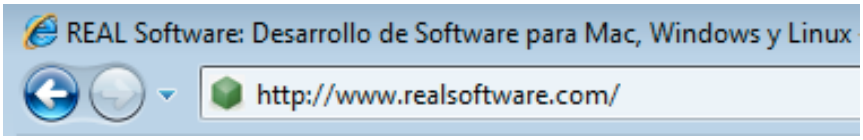
código causante del problema. Comprueba de nuevo la línea con su versión impresa. El *Tutorial* incluye una sección de *resolución de problemas* que que ayudará a resolver los errores de sintáxis.

Comenzamos

Crearás una aplicación que gestiona direcciones URL y direcciones de correo. Un URL (abreviatura, en inglés, de *Uniform Resource Locator*) es la dirección de una página web que escribes en un área de dirección del navegador web. Esta aplicación ejecutará tu aplicación de navegación web por omisión y mostrará la página web correspondiente a la dirección que hayas introducido.

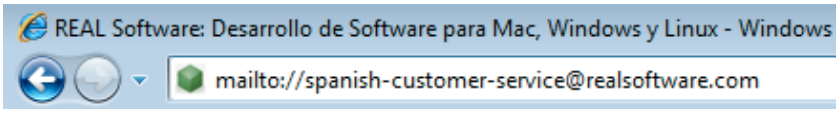
La Figura 1 muestra un URL en el área de Dirección del navegador de internet.

Figura 1. Un URL introducido en un navegador.



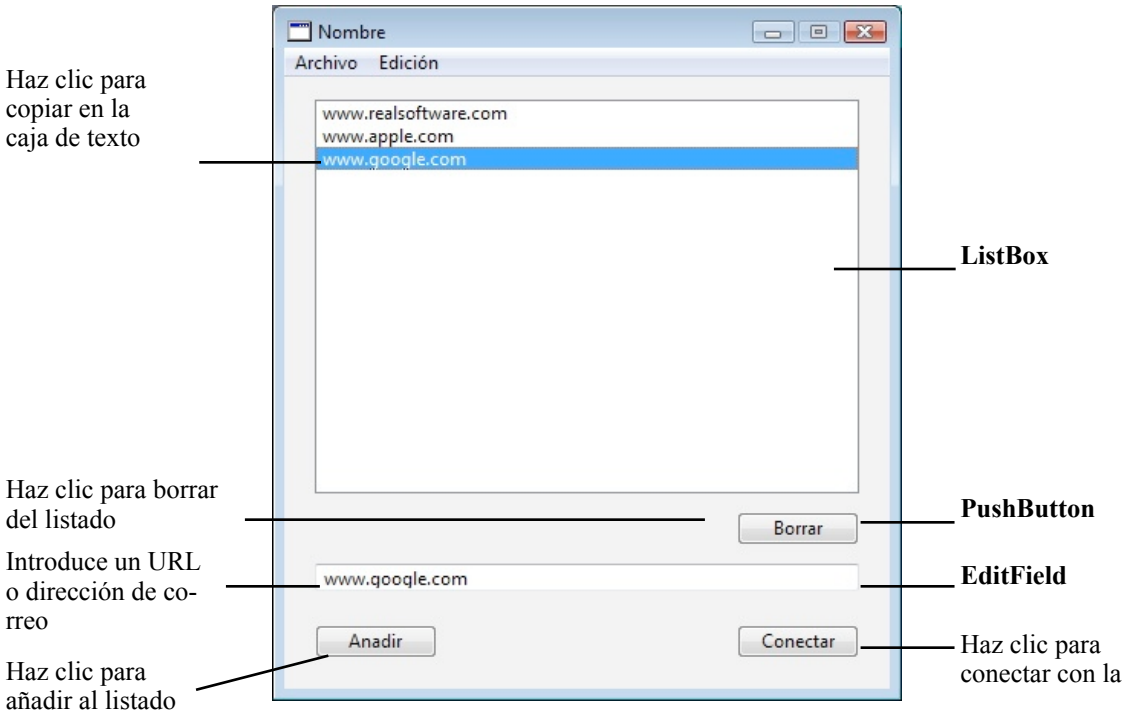
Si introduces una dirección de correo en vez de un URL, entonces la aplicación ejecutará tu aplicación de correo electrónico, creará un nuevo mensaje en blanco e introducirá la dirección en el campo "Para". La Figura 2 muestra cómo puedes introducir una dirección de correo en el área de Dirección de un navegador.

Figura 2. Una dirección de correo electrónico introducida en el área Dirección del navegador.


REAL Software: Desarrollo de Software para Mac, Windows y Linux - Windows
mailto://spanish-customer-service@realsoftware.com

Cuando hayas terminado, la aplicación tendrá el aspecto mostrado en la Figura 3. En la Figura 3, las etiquetas en negrita indican los tipos de controles que se han utilizado para crear la aplicación. Las funciones de los controles se describen en texto sencillo.

Figura 3. La aplicación de Gestión de URL una vez completada.



Utilizas el EditField para introducir el URL o la dirección de correo que quieras, haciendo clic a continuación sobre el botón Conectar. Para guardarlo en el listado, debes hacer clic en el botón Añadir. Para seleccionar un URL en el ListBox, elige el elemento en el listado y haz clic sobre Conectar. Si ya no necesitas un URL, elígelo en el listado y haz clic sobre el botón Borrar.

La aplicación utiliza tres tipos de controles REALbasic para hacer la mayor parte del trabajo:

- Un **ListBox** es el tipo de control que contiene listados que puedes recorrer. Puede contener tanto listados de una columna como de múltiples columnas, además de admitir desplazamientos en sentido vertical y horizontal.
- Un **EditField** es el tipo de control que contiene texto. Puede utilizarse para contener tanto una línea de texto (tal y como se usa aquí) o bien como un editor de textos de múltiples líneas.
- Un **PushButton** es el tipo de botón estándar. Es el utilizado con frecuencia para iniciar una acción (tal y como ocurre aquí).

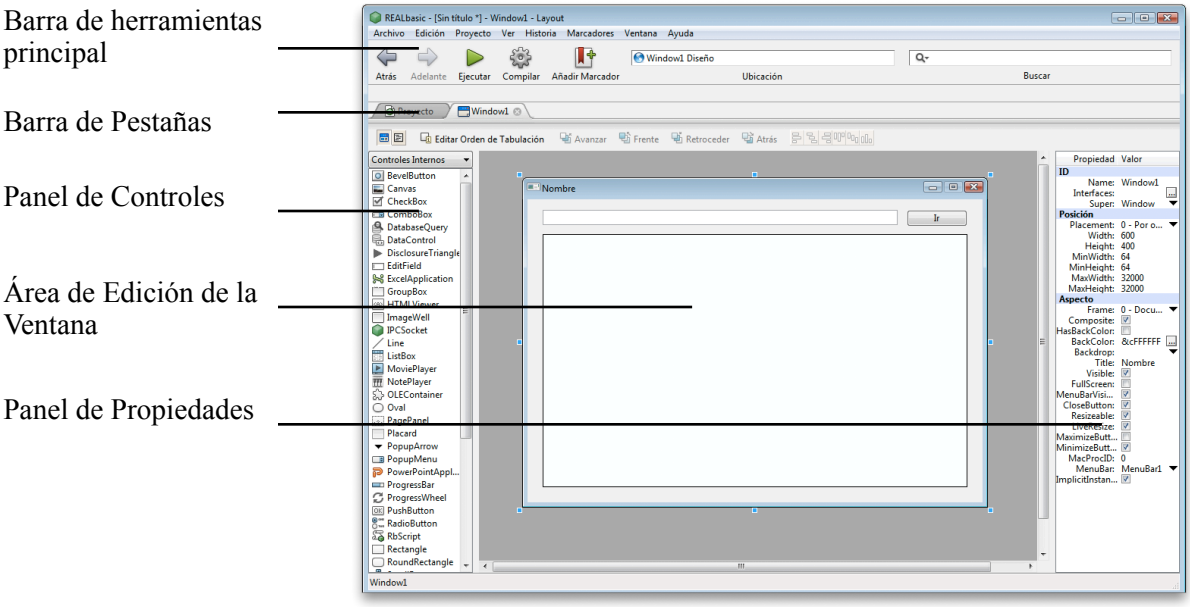
Creando la Interfaz

Si aun no lo has hecho todavía, haz doble clic sobre el icono de aplicación de REALbasic. En unos segundos aparecerá la ventana del Entorno de Desarrollo Integrado (IDE) de REALbasic. En el caso de que aparezca la ventana de Bienvenida, ciérrala de modo que puedas acceder al IDE de REALbasic.

El IDE utiliza una interfaz de pestañas para mostrar múltiples editores en la misma ventana. REALbasic muestra el Editor de Diseño de Ventana como el panel principal en el IDE. Puedes cambiar entre los diferentes editores haciendo clic sobre la barra de Pestañas. Advierte que ya está visible una segunda pestaña correspondiente al Editor de Proyecto. Iremos al Editor de Proyecto posteriormente. Tu primera tarea consiste en crear la interfaz en el Editor de Diseño de Ventana.

La Figura 4 muestra el editor correspondiente a Window1.

Figura 4. El Editor de Diseño de Ventana para Window1.



El Editor de Diseño de Ventana está dividido en tres paneles. El panel de la derecha es el panel de Propiedades y muestra las propiedades del elemento que esté seleccionado en ese momento sobre el área de Edición de la Ventana. En la Figura 4 está seleccionada la ventana, de modo que el panel Propiedades muestra las propiedades correspondientes a la ventana.

El panel central es el área de Edición de la Ventana. Aquí es donde diseñas la ventana. En la Figura 4, la ventana está vacía dado que

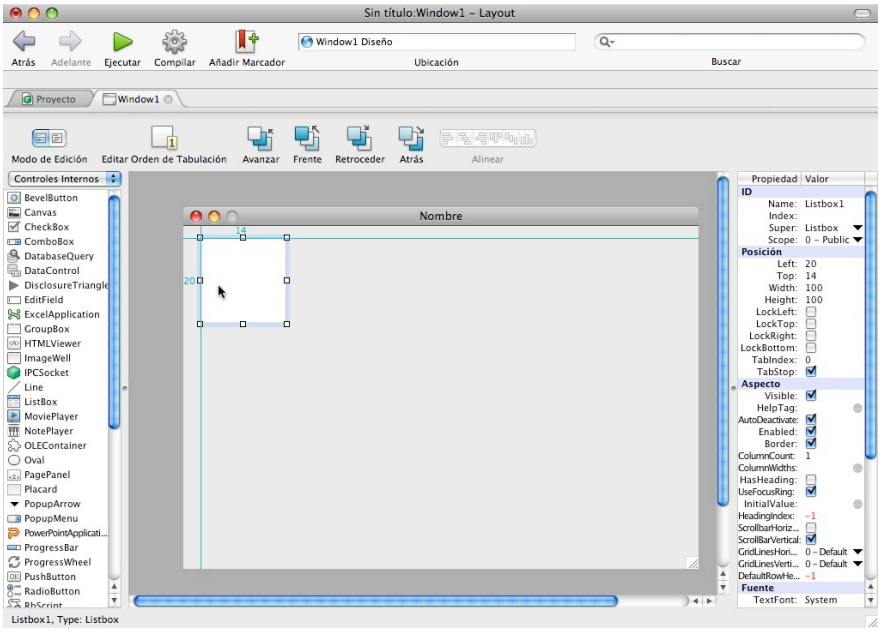
aun no has añadido ningún contenido sobre ella. Sólo muestra la ventana de documento por omisión.

El panel de la izquierda es el listado de los controles que puedes añadir a una ventana. Sobre el listado se encuentra un menú desplegable y que se utiliza para filtrar el tipo de controles mostrados en el listado. Cuando se elige la opción "Controles incorporados" en dicho menú, se mostrará el listado correspondiente a los elementos de interfaz de serie en REALbasic; como por ejemplo ListBox, EditField, PushButton y muchos otros. Añades un control a una ventana arrastrándolo desde el listado hacia la posición que deseas que ocupe sobre la ventana.

1 En el panel Controles, mantén pulsado el botón del ratón sobre el elemento ListaBox y arrástralo a la esquina superior izquierda de la ventana en el área de Edición de la Ventana.

A medida que lo arrastres cerca de la esquina, aparecerán unas guías de alineamiento, tal y como se muestra en la Figura 5. Es probable que debas soltar el ListBox cerca del área correcta y que debas cogerlo de nuevo para que puedas ver las guías de alineación.

Figura 5. Las guías de alineación te ayudan a situar el ListBox y alinearlo con otros controles.

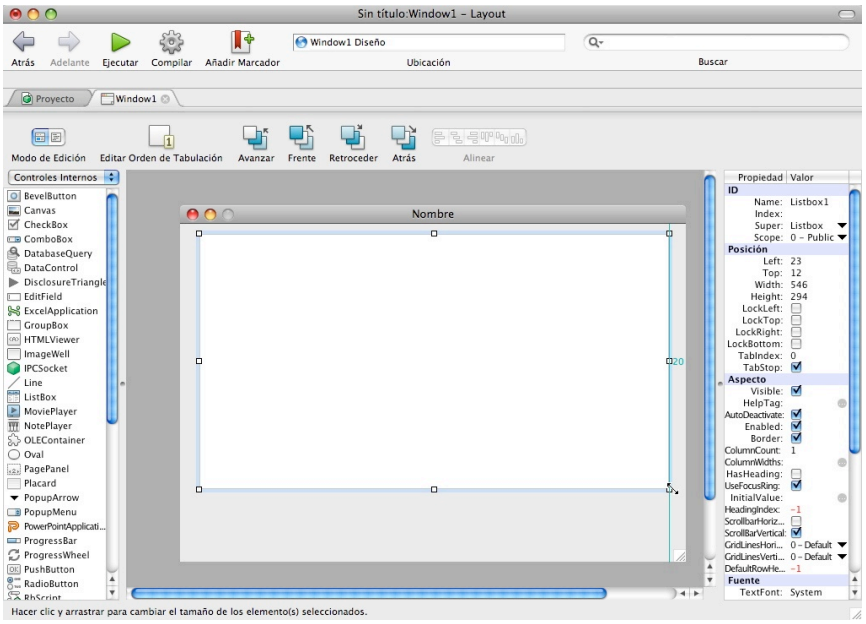


2 Suelta el botón del ratón.

Aparecerá un cuadrado de ListBox en el Editor de Ventana.

3 Coge el manejador de tamaño situado en la esquina inferior derecha del ListBox y arrástralo diagonalmente para aumentar el tamaño del ListBox. Cuando aparezca la guía de alineación en la derecha, suelta el botón del ratón, tal y como se muestra en la Figura 6.

Figura 6. Aumentando el tamaño del ListBox.

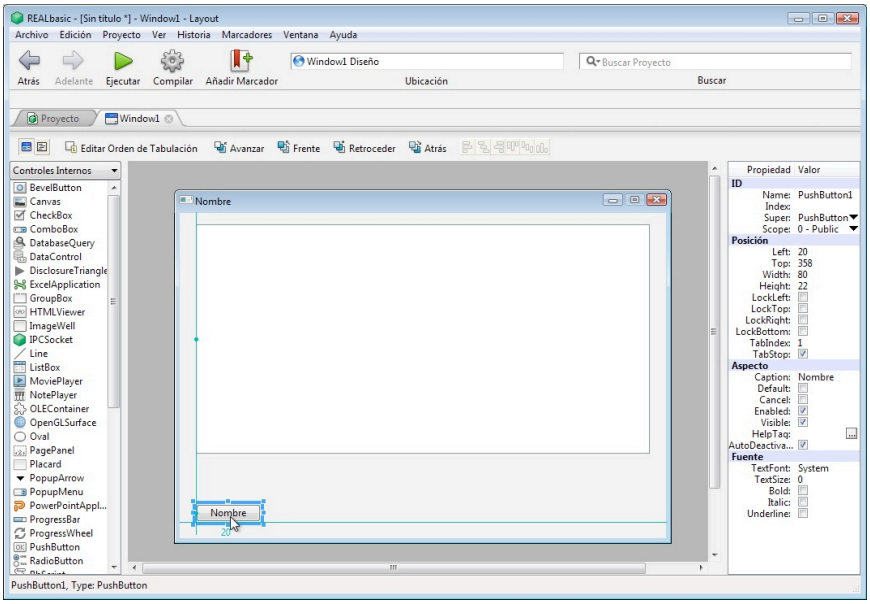


4 Céntralo sobre el área superior de la ventana, tal y como se muestra en la Figura 6.

5 A continuación, arrastra un control PushButton desde el panel de Controles hacia la esquina inferior izquierda de la ventana (donde se encuentra el botón Añadir en la Figura 3).

A medida que lo arrastras cerca de la esquina de la izquierda aparecerá una guía de alineación vertical. Utiliza la guía de alineación vertical para situar el PushButton, tal y como se muestra en la Figura 7.

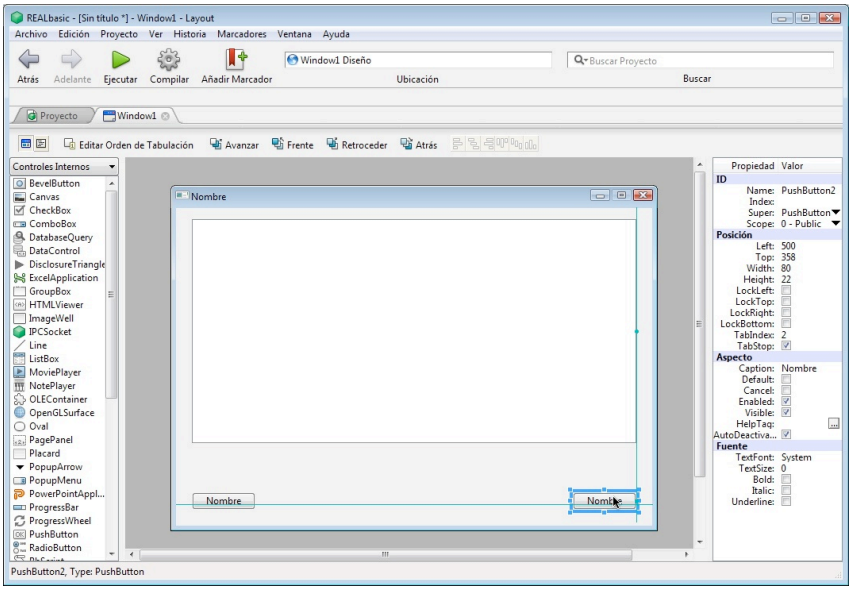
Figura 7. Alineando el PushButton Añadir.



6 Con el PushButton seleccionado, pulsa Ctrl - D (Windows y Linux) o Comando - D (Macintosh), o bien selecciona Edición > Duplicar para crear una copia del PushButton. Muévelo a la derecha del botón anterior y alinéalo con la esquina derecha del ListBox.

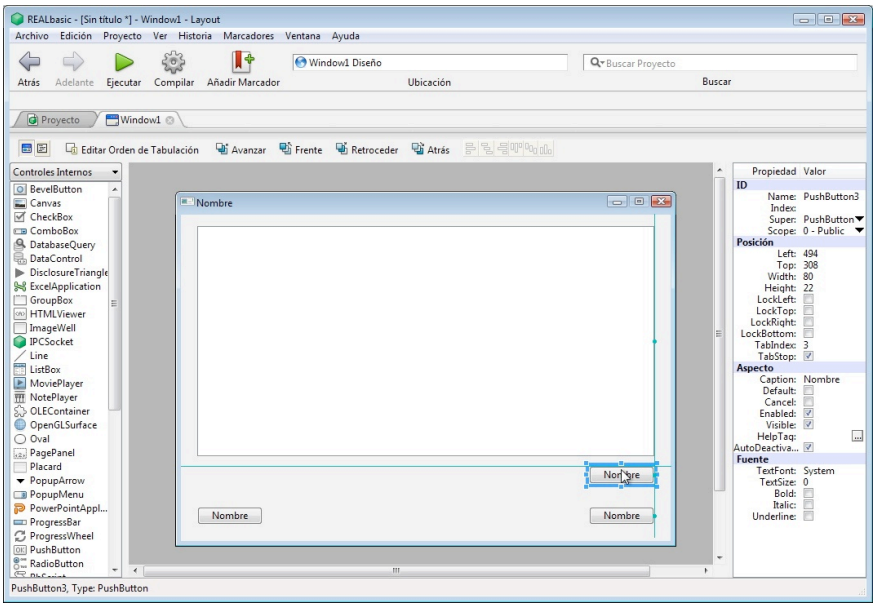
Cuando alcances la posición aproximada, aparecerán las guías de alineación vertical y horizontal. Alinea las líneas base correspondientes a los dos encabezados de ambos PushButton, tal y como se muestra en la Figura 8.

Figura 8. Alineando los PushButton Añadir y Conectar.



7 Con este PushButton seleccionado, pulsa Ctrl - D (Windows y Linux) o Comando - D (Macintosh) o bien selecciona Edición > Duplicar para crear una copia del PushButton, y arrastra el PushButton verticalmente hacia arriba, justo bajo el ListBox. Suelta el botón del ratón cuando aparezca la guía de alineación horizontal.

Figura 9. Añadiendo el botón Borrar.

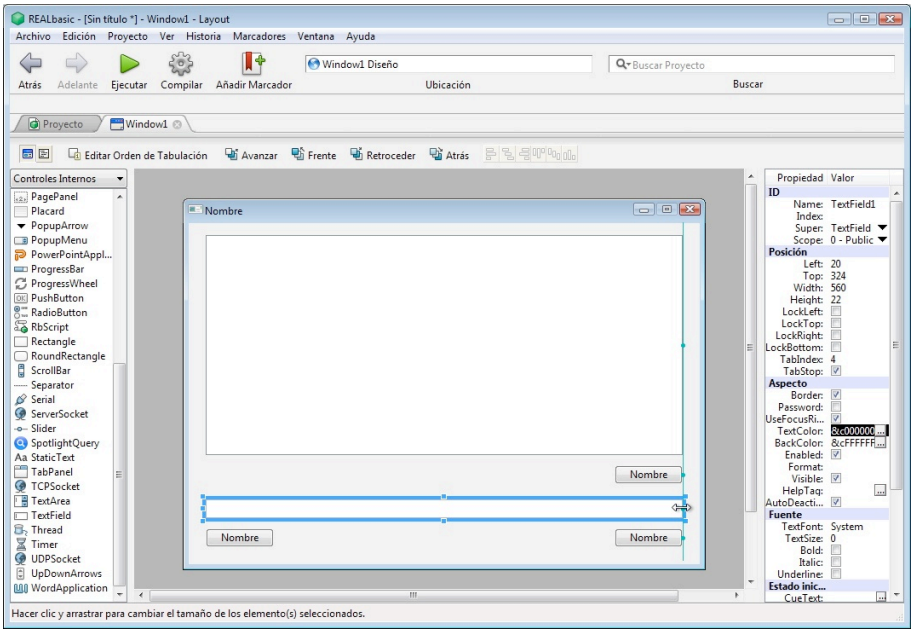


8 En el panel Controles, mantén pulsado el botón del ratón sobre el EditField y arrástralo hacia su posición en la ventana, a mitad de camino entre los botones Borrar y Conectar (observa la Figura 10).

9 Alinea el margen izquierdo del EditField con el margen izquierdo del ListBox.

10 Utiliza el manejador de selección central para acortar el EditField de modo que su margen derecho se alinee con el margen derecho del ListBox.

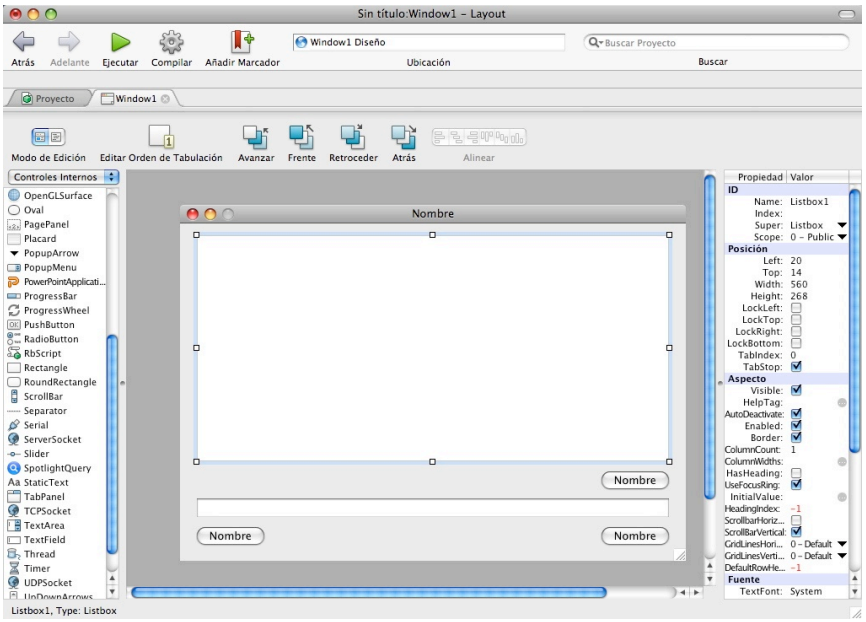
Figura 10. Alineando el margen derecho del EditField con el ListBox y los PushButtons.



Si observas que el PushButton superior, el EditField y la fila inferior de los PushButton no tienen una distribución armoniosa, puedes cambiar la posición de un control haciendo clic sobre él y moviéndolo pixel a pixel utilizando las teclas de cursor Arriba y Abajo.

¡Ya has completado la interfaz de la aplicación! Debería tener el aspecto mostrado en la Figura 11.

Figura 11. La interfaz completada en el IDE de REALbasic.



Guarda el proyecto antes de que sigamos avanzando.

- Selecciona Archivo > Guardar (Ctrl - S o Comando - S) y nómbralo Gestor URL.

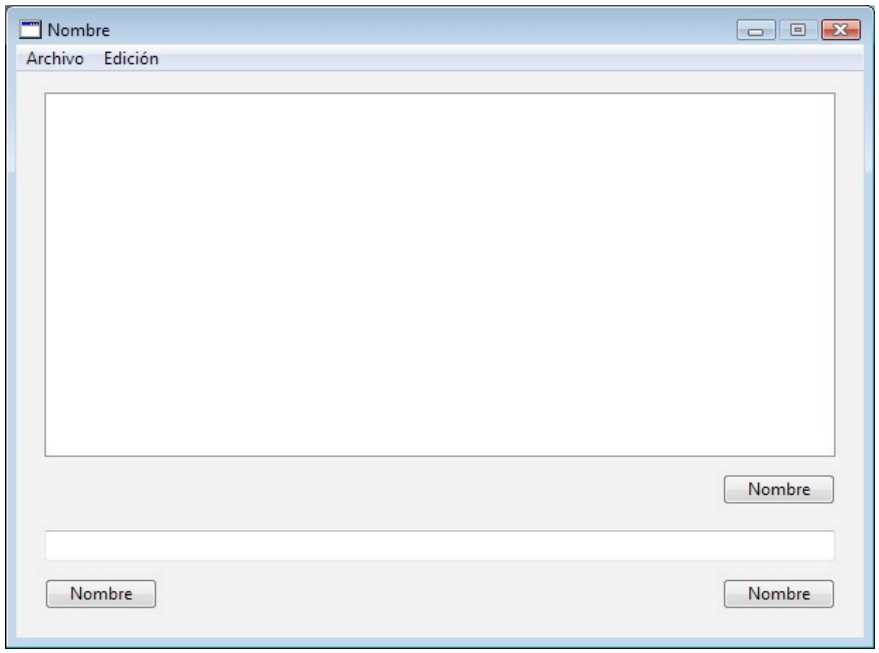
De hecho, en este punto ya puedes probar la aplicación. Por supuesto no hará nada dado que no hemos indicado a ninguno de los elementos de la interfaz qué es lo que deben hacer.

1 Sólo por diversión, haz clic sobre el botón Ejecutar en la Barra de Herramientas Principal.

REALbasic compilará la aplicación y abrirá su propia ventana.

La primera versión de la aplicación debería tener este aspecto:

Figura 12. La primera versión de la aplicación.



En este estado, los PushButton funcionan (es decir, puedes hacer clic sobre ellos y estos se seleccionarán), aunque no harán nada dado que no hemos indicado a los PushButton qué deben hacer cuando se haga clic sobre ellos. Puedes introducir texto en el EditField (aunque no servirá de mucho dado que no hay instrucciones para procesar dicho texto). Y el ListBox está preparado para mostrar los elementos, pero aun no tenemos ningún modo de introducir texto en el ListBox todavía.

Advierte que la aplicación también tiene los menús Archivo y Edición. En Windows y Linux, el menú Archivo tiene un elemento: Salir,

y que se encarga de salir de la aplicación y volver al IDE de REALbasic. En el Mac OS X, el elemento de menú Salir se encuentra en el menú de aplicación en vez de en el menú Archivo.

El menú Edición contiene los elementos estándar Deshacer, Cortar, Copiar, Pegar, Borrar y Seleccionar Todo. Inicialmente estos elementos se encuentran desactivados, pero puedes activarlos introduciendo y seleccionando texto en el EditField.

2 Escribe algún texto en el EditField y despliega a continuación el menú Edición.

Advierte que ahora aparece activado el elemento de menú Seleccionar Todo del menú Edición.

3 Selecciona Edición > Seleccionar Todo y selecciona a continuación Edición > Cortar.

Cuando cortas texto al Portapapeles se activa el elemento de menú Pegar. Todos los elementos estándar del menú Edición funcionan sobre el texto sin que se necesario que los programes.

Ahora necesitamos volver al IDE de REALbasic para hacer que esta aplicación sea operativa.

4 En la aplicación Gestor URL, selecciona Archivo > Salir en Windows o Linux (o MiAplicación.debug > Salir, en Mac OS X) para volver al IDE de REALbasic. En Windows y Linux también puedes salir de la aplicación haciendo clic sobre el elemento de cierre de la ventana.

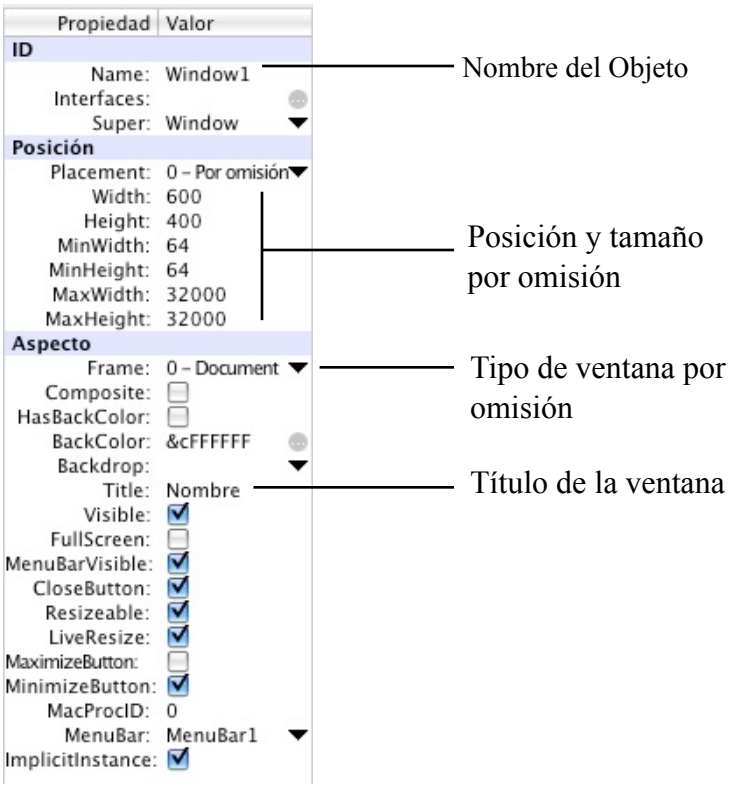
Asignar Nombres y Etiquetas Significativos a los Objetos

Ya habrás advertido que todos los PushButton utilizan la etiqueta por omisión “Sin título”, También tienen los nombres por omisión como PushButton1, PushButton2, etc. Antes de que sigamos avanzando en el proyecto, deberíamos dar a los objetos nombres y etiquetas significativos. En el código de REALbasic utilizarás el nombre de los objetos para referirte a ellos y, por supuesto, las etiquetas son los textos mostrados al usuario. Al inicio del proyecto deberías dar un nombre significativo a cada uno de los objetos que vayas a referenciar mediante código.

1 Clic sobre la superficie de la ventana en el Editor de Ventana, y no sobre cualquiera de los controles contenidos en la misma, y echa un vistazo a continuación al panel de Propiedades.

La ventana tiene el nombre por omisión "Window1", y que aparece en la primera línea. Si muestra algún otro nombre, entonces es porque habrás hecho clic sobre alguno de los controles de la ventana en vez de haber hecho clic sobre la superficie de la ventana propiamente dicha.

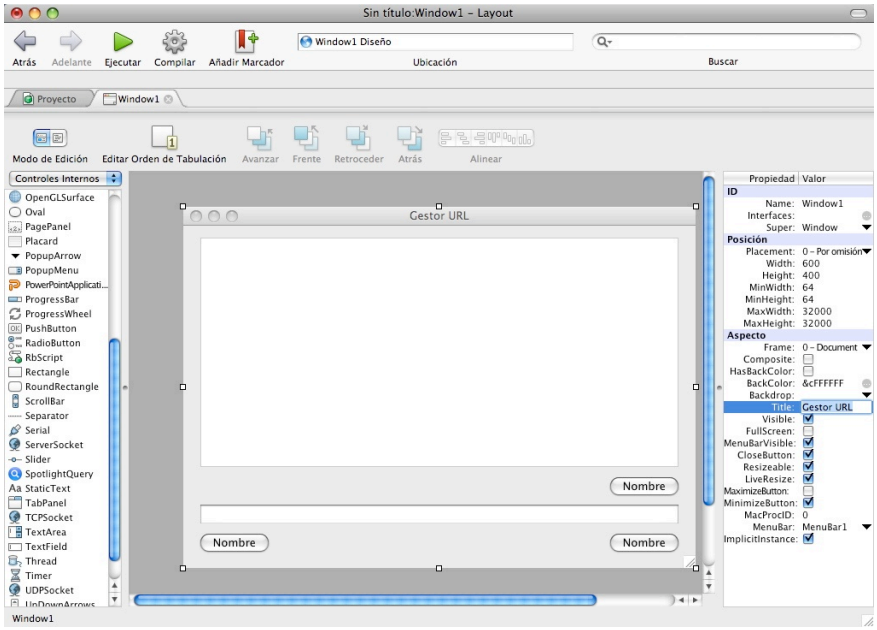
Figura 13. Panel de Propiedades de Window1.



El texto que aparece en la barra de Título en la ventana se corresponde con la propiedad Title. La propiedad Name es el nombre de la ventana que utilizarás desde el código para referirte a dicho objeto.

2 Selecciona el texto por omisión de la propiedad Title, "Sin título", en el panel de Propiedades y sustitúyelo por el texto **Gestor URL**, pulsando a continuación la tecla Retorno. Cuando pulses la tecla Retorno, aparecerá el nuevo título en la barra de Título del Editor de Ventana así como en el panel Propiedades.

Figura 14. Cambiando la propiedad Title de Window1.



De igual modo, tenemos que sustituir los nombres y etiquetas por omisión de los controles utilizados en la ventana.

3 En el Editor de la Ventana, selecciona el botón Sintitulo situado en la esquina inferior izquierda haciendo clic sobre él.

Advierte que el panel Propiedades cambia para mostrar las propiedades de este PushButton. Los tres PushButton tienen los nombres PushButton1, PushButton2 y PushButton3. Han sido nombrados atendiendo al orden en el que han sido creados. Nunca recordaríamos cual es cada uno de modo que lo mejor es renombrarlos al mismo tiempo que introducimos sus etiquetas.

4 Cambia la propiedad Name de este PushButton a AnadirURL (en REALbasic no puedes utilizar acentos ni caracteres como la "ñ" o la "ç" en el nombre de los objetos) y la propiedad Caption a **Añadir**. Pulsa la tecla Retorno después de haber introducido cada entrada para guardar el valor de cada nueva propiedad.

Advierte que el texto de la propiedad Caption sustituye de inmediato a "Sin título" en el Editor de la Ventana cuando pulsas la tecla Retorno. Una vez que hayas finalizado, el panel de Propiedades correspondiente al PushButton AnadirURL debería tener el siguiente aspecto:

Figura 15. El panel de Propiedades para el botón Añadir.

5 Haz clic sobre el botón Sin título en la esquina inferior derecha para seleccionarlo. Se trata del PushButton2. Utiliza el panel Propiedades para cambiar su nombre a **ConectarURL** y su propiedad Caption a **Conectar**.

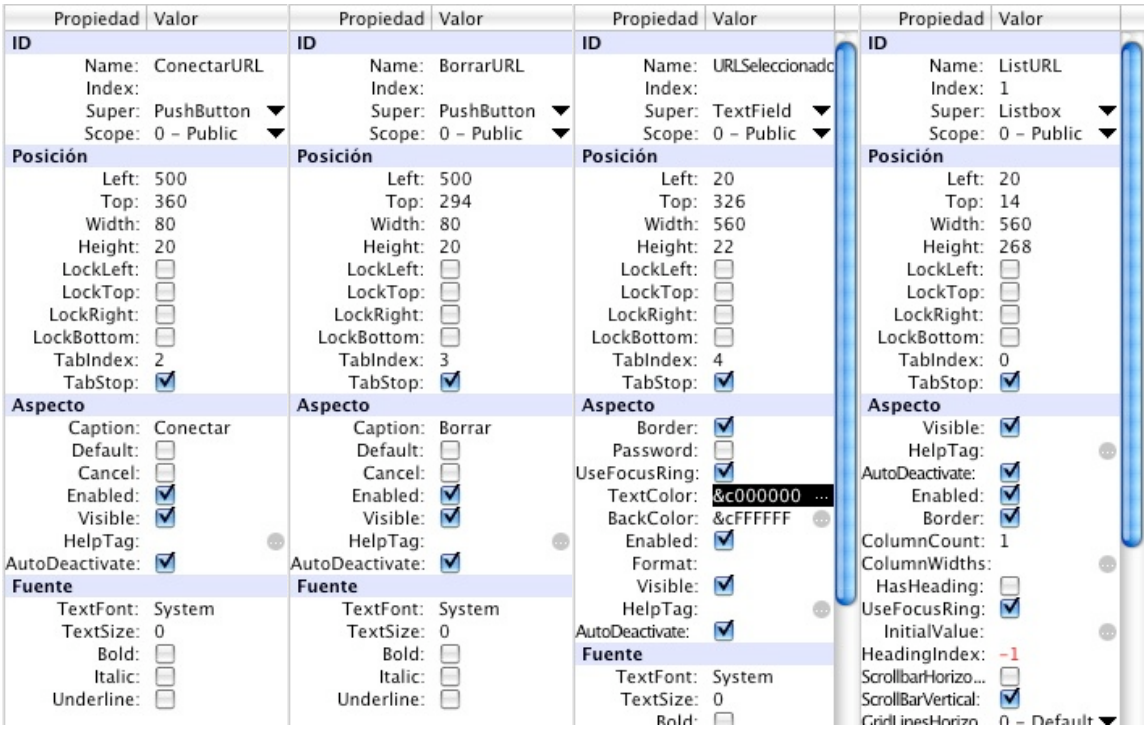
6 Haz clic en el PushButton Sin título situado entre el ListBox y el EditField. Cambia su propiedad Name a **BorrarURL** y su propiedad Caption a **Borrar**.

7 Haz clic sobre el EditField y cambia su propiedad Name a **URLSeleccionado**.

8 Haz clic sobre el ListBox y cambia su propiedad Name a **ListURL**.

Ya hemos hecho lo correcto. Los paneles de Propiedades para estos objetos deberían tener el aspecto mostrado en la Figura 16.

Figura 16. Los paneles de Propiedades para el resto de los controles.



Botón Conectar Botón Borrar TextField ListBox

Comprueba los cambios para asegurarte de que los elementos tienen los nombres correctos. Si existe un error ortográfico en la propiedad Name, no funcionará el código que se refiera a dicho elemento.

En el Editor de la Ventana, los tres botones deberían aparecer con el siguiente aspecto:

Figura 17. Los tres controles PushButton una vez les hemos cambiado su nombre.

9 Selecciona Archivo > Guardar para guardar los cambios.

10 Haz clic en el botón Ejecutar de la Barra Principal de Herramientas para probar la aplicación.

No hará mucho más en comparación con la anterior versión, pero por lo menos ahora todos los elementos de la interfaz muestran las etiquetas correctas.

11 Cuando hayas terminado de probar la aplicación, selecciona Archivo > Salir (en Windows y Linux) o bien MiAplicacion.debug > Salir (en Mac OS X) para regresar al IDE de REALbasic. En Windows y Linux también puedes salir de la aplicación haciendo clic sobre el elemento de cierre de la ventana.

Asegúrate de salir de la aplicación compilada antes de continuar con tu trabajo en el IDE de REALbasic. Si no lo has hecho, cierra la pestaña "Ejecutar" en la barra Pestañas.

Logra que el Gestor URL haga algo

Ahora que la interfaz está diseñada y tras los oportunos retoques, es el momento de lograr que los controles hagan su trabajo. Empezaremos con el botón Conectar. Especificas la acción realizada por un botón escribiendo algo de código que pasa a formar parte de dicho botón.

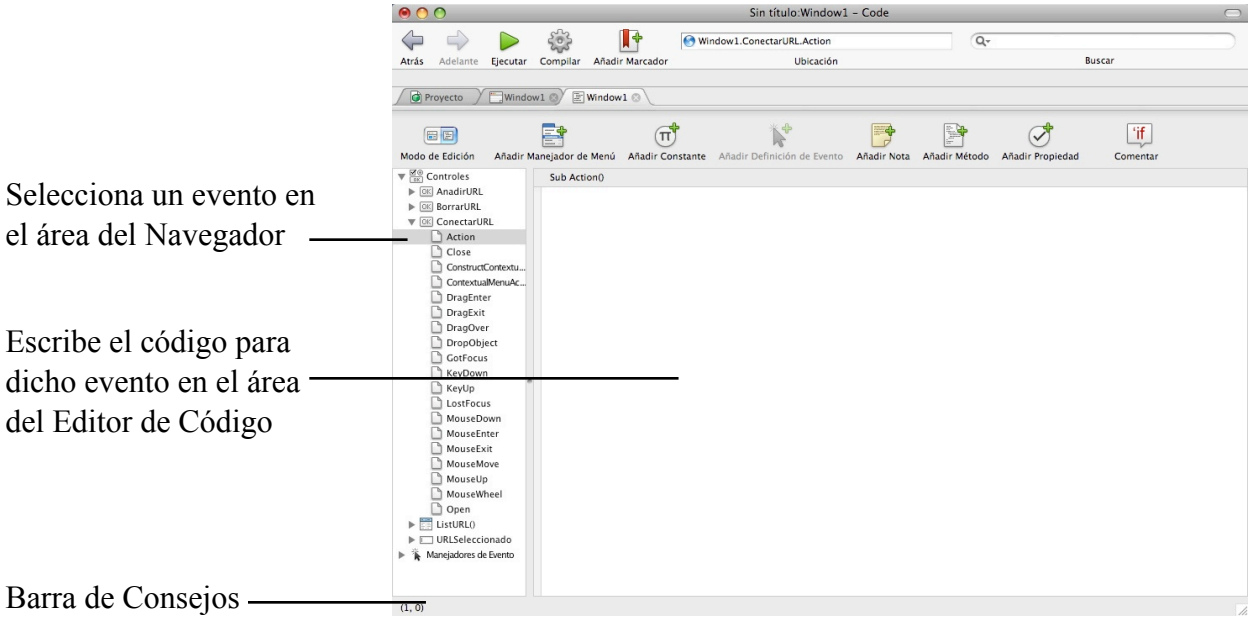
1 En el Editor de la Ventana, haz doble clic sobre el botón Conectar.

Aparecerá el Editor de Código correspondiente a Window1. Aquí es donde escribes el código para la ventana y los controles contenidos en dicha ventana.

En el margen izquierdo se encuentra el área del navegador que lista todos los controles que se han añadido a la ventana, entre otras cosas. (Para el *Tutorial* sólo necesitamos trabajar con los controles.) En el margen derecho se encuentra el área para la edición del código. Dicha área contiene el código para el elemento que se encuentre seleccionado en ese momento en el navegador.

Justo ahora, está seleccionado el elemento "Action" del botón Conetar. Está destacado en el área del Navegador y el área de encabezado sobre el editor te indica el nombre del evento.

Figura 18. El Editor de Código para Window1.



Para conseguir que el botón Conectar haga algo, necesitamos escribir algo de código que se ejecute sólo cuando el usuario haga clic sobre el botón. Afortunadamente, la propia aplicación REALbasic se encarga de monitorizar toda la actividad de la interfaz del usuario durante el funcionamiento de la aplicación y sabe cuando ocurre.

Para que el botón Conectar funcione, necesitamos escribir las instrucciones que conecta al usuario con el sitio web que se haya introducido en el EditField.

En el área del Navegador correspondiente al Editor de Código, verás un listado de los eventos que REALbasic monitoriza de forma constante durante el funcionamiento de la aplicación. Por ejemplo, se lanzan los eventos MouseEnter y MouseExit cuando el apuntador del ratón entra en la región del PushButton y abandona dicha región, respectivamente.

El evento que necesitamos es el evento “Action”. Este evento tiene lugar cuando el usuario hace clic sobre el botón.

En el margen derecho del divisor puedes escribir el código que se ejecuta automáticamente cuando el usuario hace clic sobre el botón. (Advierte que la primera línea, “Sub Action()”, indica a qué evento corresponde el código). La instrucción para abrir el sitio web en el navegador del usuario (o para abrir la aplicación de correo electrónico) es simple. La instrucción ShowURL y su sintaxis es:

```
ShowURL string
```

Donde *string* es el URL (o dirección de correo electrónico).

Cuando escribas “ShowURL”, aparecerá la sintaxis para dicho comando en la barra de Consejos. Consultando la barra de Consejos evitarás la necesidad de tener que consultar la sintaxis de un comando en la Referencia del Lenguaje en línea cada vez que necesites ayuda.

Figura 19. La barra de Consejos mostrando la sintaxis para ShowURL.

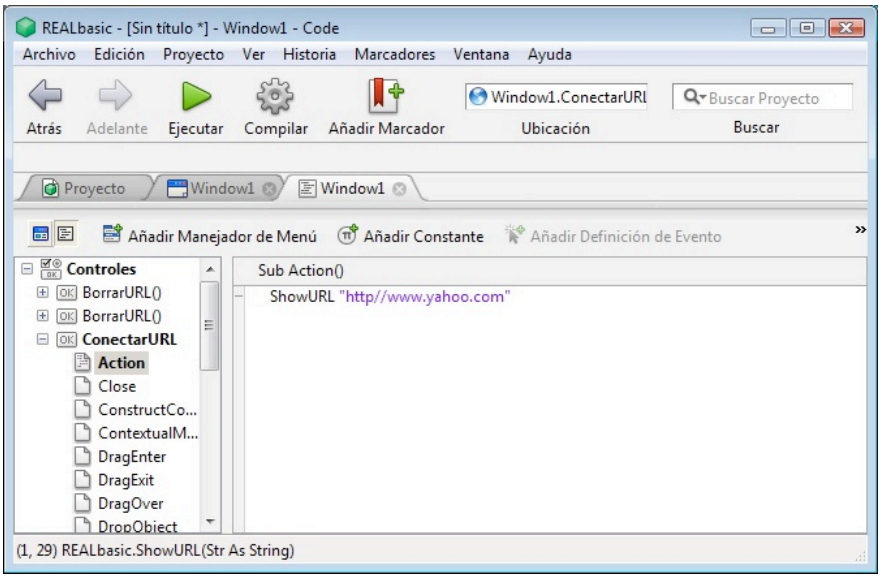
```
(1, 7) REALbasic.ShowURL(Str As String)
```

2 Para probar este botón, introduce la siguiente línea en el evento Action de ConectarURL:

```
ShowURL “http://www.yahoo.com”
```

Asegúrate de encerrar el URL entre comillas. Las comillas indican que el texto contenido es una cadena (en vez de un número o bien cualquier otro tipo de información). Tu Editor de Código debería mostrar el siguiente aspecto:

Figura 20. El código para el evento Action de ConnectURL.



3 Para probar el botón Conectar, haz clic sobre el botón Ejecutar en la Barra de Herramientas Principal.

4 Cuando aparezca la ventana de la aplicación, haz clic sobre el botón Conectar.

Tras pocos segundos se ejecutará tu navegador web para mostrarte la página web introducida. (Esto, por supuesto, asumiendo que tu ordenador tenga conexión a Internet y que tengas un navegador web por omisión.)

Figura 21. La página de inicio de Yahoo.

Por supuesto, necesitamos modificar el código de modo que el texto pasado al comando ShowURL pueda ser introducido por el usuario durante el funcionamiento de la aplicación. Cuando utilizamos ShowURL "http://www.yahoo.com", ese particular URL está "forzado".

Nota. Si obtienes un mensaje de error de sintaxis, comprueba que hayas escrito correctamente el nombre del comando ShowURL, y de haber entrecomillado el URL, así como de comprobar que se trata de un URL válido.

---

5 Selecciona la aplicación Gestor URL y selecciona Archivo > Salir en Windows o Linux (o MiAplicación.debug > Salir en Mac OS X) para volver al IDE de REALbasic.

Ahora necesitamos sustituir el texto utilizado con el comando ShowURL por el código que se refiere a los contenidos del EditField. URLSeleccionado.

Dado que el nombre del EditField es "URLSeleccionado", quizá pienses que podríamos escribir:

```
ShowURL URLSeleccionado
```

Pero eso no funcionará porque "URLSeleccionado" es el nombre del objeto propiamente dicho. Dicho objeto tiene una buena cantidad de propiedades, como su posición sobre la ventana, si emplea varias líneas de texto o sólo una, si puede aceptar texto con estilos o no, etc. Si utilizas "ShowURL URLSeleccionado", REALbasic no sabrá lo que quieres decir. Además, el comando ShowURL sólo acepta una cadena de texto, y no un control.

Cuando necesitas referirte a una de las propiedades del objeto, has de escribir el nombre del objeto seguido por un punto, seguido por el nombre de la propiedad. En otras palabras, debes utilizar esta sintaxis:

```
nombredeobjeto.nombredepropiedad
```

A veces se indica este tipo de sintaxis como notación de punto.

En este caso, el EditField se llama "URLSeleccionado" y la propiedad del EditField que queremos utilizar es su propiedad "Text". Esto significa que la siguiente expresión accede a los contenidos del EditField:

```
URLSeleccionado.Text
```

Es decir, "URLSeleccionado" es el nombre del objeto y "Text" es el nombre de la propiedad del objeto que necesitamos.

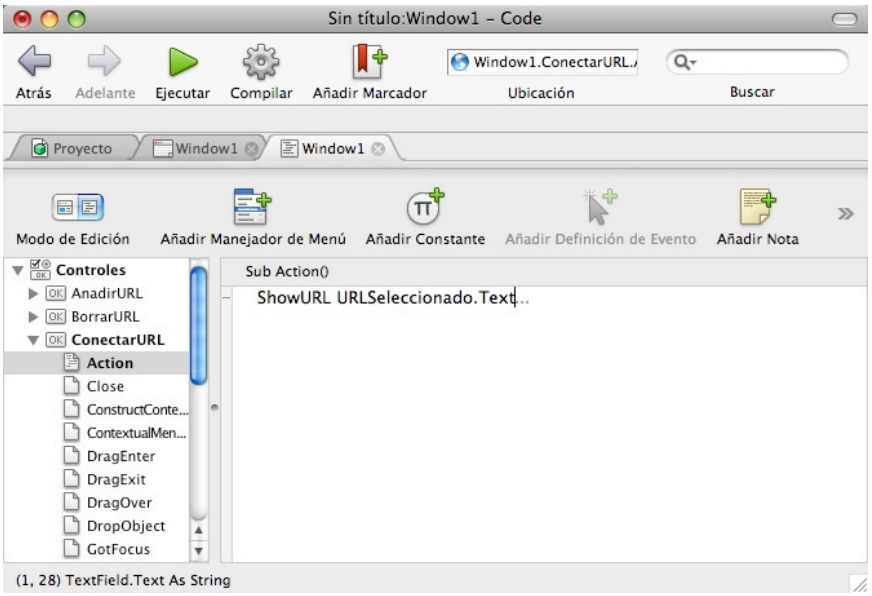
1 En el Editor de Código para el evento Action, modifica el código para que sea:

```
ShowURL URLSeleccionado.Text
```

Esta expresión URLSeleccionado.Text se refiere sólo a la propiedad de texto.

Tu Editor de Código debería tener el siguiente aspecto:

Figura 22. El código modificado para el evento Action.



Si tienes problemas introduciendo esta línea de texto, asegúrate haber salido previamente de la aplicación de prueba antes de volver al IDE de REALbasic. Sólo has de salir de la aplicación de prueba.

2 Guarda el proyecto (Archivo > Guardar) y pruébala haciendo clic sobre el botón Ejecutar en la Barra de Herramientas Principal.

La aplicación Gestor URL se abrirá en una nueva ventana.

3 Introduce un URL en el EditField, como por ejemplo **http://www.realsoftware.com** y haz clic en el botón Conectar.

Debería abrirse tu navegador por omisión para mostrar la página web correspondiente al URL introducido.

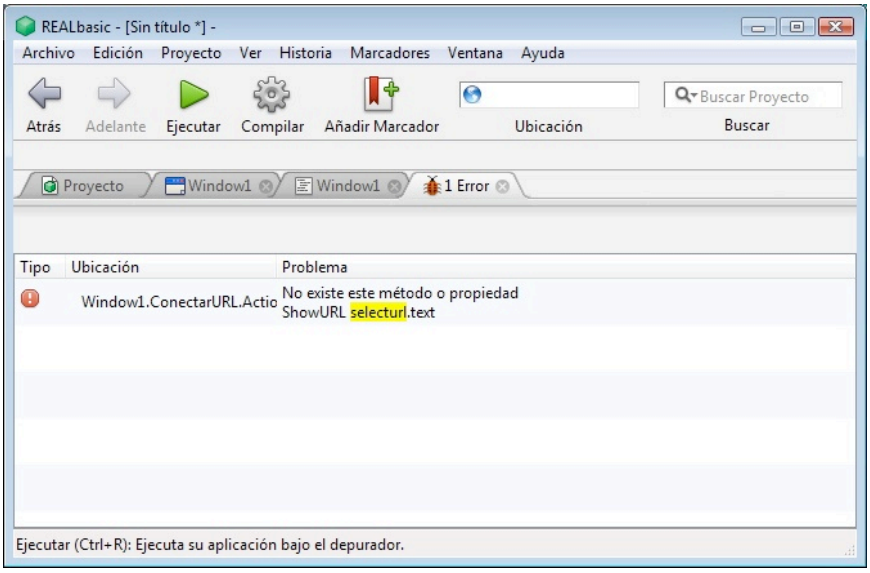
4 Cuando hayas terminado, sal del entorno de depuración (Archivo > Salir en Windows y Linux; MiAplicación.debug > Salir en Macintosh) para volver al IDE de REALbasic.

Si la Aplicación no Funciona

En el caso de que REALbasic no haya sido capaz de ejecutar la aplicación en modo de depuración, será porque no habrá reconocido alguno de los términos introducidos en el Editor de Código. Por ejemplo, si has cometido algún error al teclear “ShowURL” o

"URLSeleccionado", REALbasic se parará y te indicará el término que no reconoce. En vez de compilar y ejecutar la aplicación, abrirá una pestaña de "problemas" en el IDE. Dicha pestaña te mostrará una línea por cada uno de los problemas detectados. Por ejemplo, lo siguiente muestra un error de tecleo.

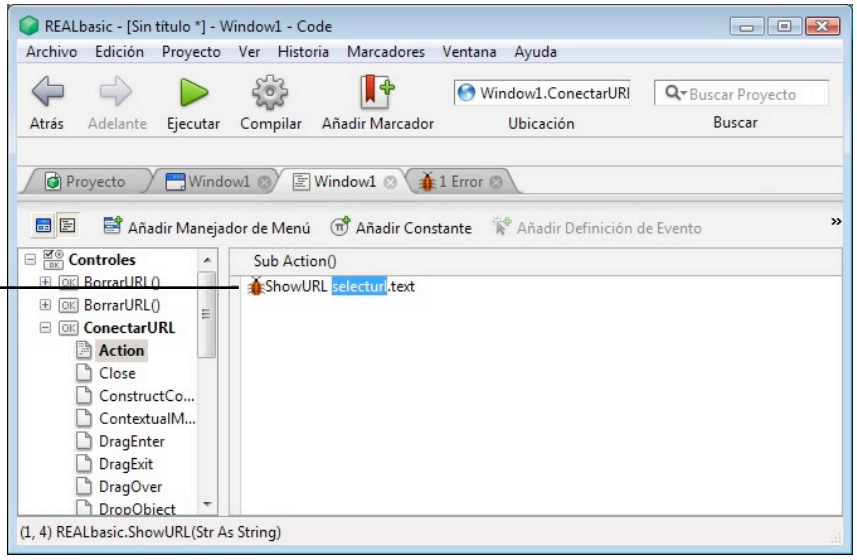
Figura 23. Un error de tecleo resulta en un término que no reconocido.



El término no reconocido aparecerá destacado. Haz doble clic sobre la línea para dirigirte al editor en el que se encuentre dicho error. En este caso, REALbasic muestra el Editor de Código correspondiente al evento Action de URLConectado, mostrado en la Figura 24. La línea que contiene el error tiene a su lado el icono de bug.

Figura 24. Un nombre de objeto mal escrito.

Línea que contiene el error



Dado que no se puede encontrar un objeto llamado "URLSelec-cion", no puede crear la aplicación. REALbasic sabe que no será capaz de averiguar qué hacer cuando el usuario haga clic sobre el botón Conectar. Asegúrate de que has nombrado los controles tal y como se ha descrito, y que te has referido a ellos utilizando sus nombres correctos en el Editor de Código. También obtendrás este error en el caso de que hayas escrito MostrarURL URLSeleccionado sin haber añadido también el nombre de la propiedad que incluye los contenidos del EditField.

Otra posibilidad es que hayas utilizado el nombre correcto en el Editor de Código pero que no hayas cambiado el nombre del EditField utilizando el panel de Propiedades tal y como se muestra en la Figura 16. En dicho caso, la instrucción del Editor de Código que se muestra en la Figura 22 rehusará funcionar, dado que en la aplicación no existirá ningún objeto con el nombre "URLSeleccio-nado". Si obtienes un mensaje de error, empieza por revisar el término destacado.

A continuación haremos que el resto de los controles realicen su trabajo.

El Botón Añadir

Se supone que el botón Añadir ha de tomar el texto del EditField y añadirlo al final de la lista en el ListBox. Es sencillo.

1. En el caso de que no esté abierto el Editor de Código correspondiente a la ventana, haz doble clic sobre el botón Añadir en la ventana (si no está visible el Editor de la Ventana, haz clic en la pestaña correspondiente a Window1 en la barra Pestañas).

2. Introduce el siguiente código en el evento Action del botón Añadir:

```
ListadoURL.AddRow URLSeleccionado.Text
```

La primera parte de la expresión, ListadoURL.Addrow llama a un método interno perteneciente al ListBox. El método AddRow es un comando que añade una fila de texto al final del listado disponible en el ListBox. No es ninguna sorpresa que requiera que le pasemos el texto del nuevo elemento. Ya sabemos que URLSeleccionado.Text obtiene los contenidos del EditField, de modo que esto es lo que usaremos.

Nota. Un método es un comando que realiza una acción. Técnicamente, ShowURL es un método global porque no está asociado a un objeto en particular. Puede llamarse desde cualquier objeto que pueda llamar a un método. Nosotros nos hemos limitado a invocarlo desde un PushButton. (Por ejemplo, podríamos diseñar la aplicación de modo que ShowURL fuese invocado cuando el usuario seleccionase un elemento del menú en vez de cuando hiciese clic sobre un botón.)

Al igual que lo objetos tienen propiedades (como su nombre, tamaño, posición y etiqueta), también tienen sus propios métodos. AddRow también es un método, si bien en este caso "pertenece" sólo a los controles ListBox. Ofrece una acción especializada que sólo tiene sentido cuando se aplica a los listados en los controles ListBox.

El botón Borrar

El botón Borrar elimina el elemento seleccionado del ListBox. También es bastante simple.

1. En el Editor de Código, expande el elemento URLBorrado y selecciona el elemento Action.
2. Introduce la siguiente línea de código:

```
ListadoURL.RemoveRow ListadoURL.ListIndex
```

En este caso, estamos invocando el método RemoveRow incorporado en el control ListBox. En vez de texto, en este caso necesita el número de la fila (línea) a borrar. La propiedad ListIndex contiene dicho número, de modo que pasamos ese número al método RemoveRow.

Nota. Si sitúas el punto de inserción en el término RemoveRow, la barra de Consejos indica que RemoveRow requiere el paso de un entero.

```
(1, 16) Listbox.RemoveRow(Index As Integer)
```

El control ListBox

La tarea del ListBox consiste en copiar el item seleccionado por el usuario en el EditField, de modo que el usuario pueda conectar con dicha URL o dirección de correo electrónico. Puedes hacer dicha operación con facilidad escribiendo una *instrucción de asignación* que se ejeutará cada vez que el usuario seleccione un elemento en el listado.

1. Expande el elemento ListadoURL en el área del navegador correspondiente al Editor de Código y selecciona el manejador de evento correspondiente a Change.
2. Introduce el siguiente código en el evento Change:

```
URLSeleccionado.Text = ListadoURL.Text
```

La propiedad Text del ListBox contiene el texto del elemento seleccionado. La propiedad Text del URLSeleccionado contiene el texto

que se está mostrando. Esta instrucción de asignación copia dicho texto en la propiedad Text del EditField. La propiedad Text almacena el texto que está siendo mostrado por el EditField.

Probando la aplicación

Estas son las líneas básicas de la aplicación. Ahora es el momento de probar todas estas características.

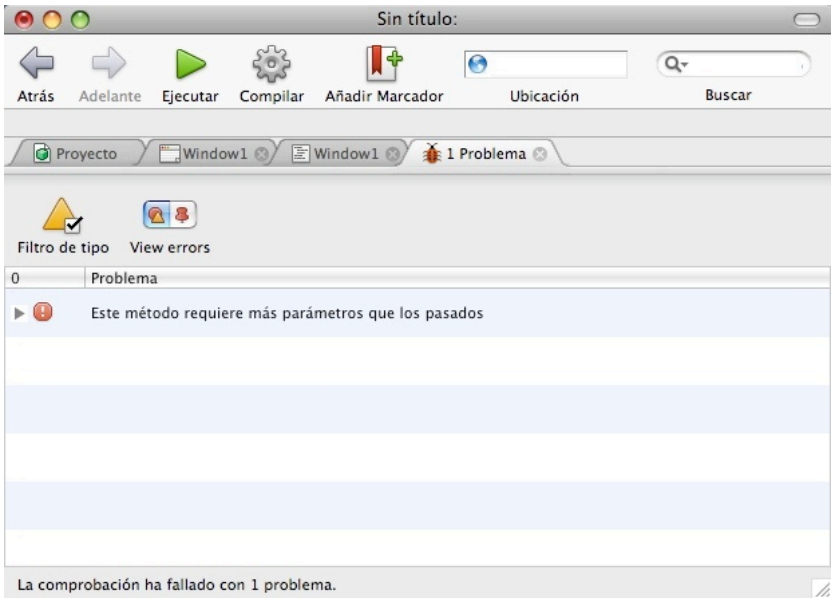
Antes de que ejecutes tu proyecto de nuevo, puedes pedir a REALbasic que compruebe el proyecto para detectar errores o bien formas de optimizar la aplicación. Para ello utilizas el comando Analizar Proyecto. La característica Analizar Proyecto utiliza el compilador de REALbasic para revisar todo el código e identificar cualquier error del código. También ofrece sugerencias sobre cómo mejorar la aplicación.

1 Selecciona Proyecto > Analizar Proyecto.

Si no encuentra problemas, mostrará el mensaje “Comprobación pasada; no se han encontrado problemas” en la barra de Consejos y *no* abrirá el panel de “Problemas”.

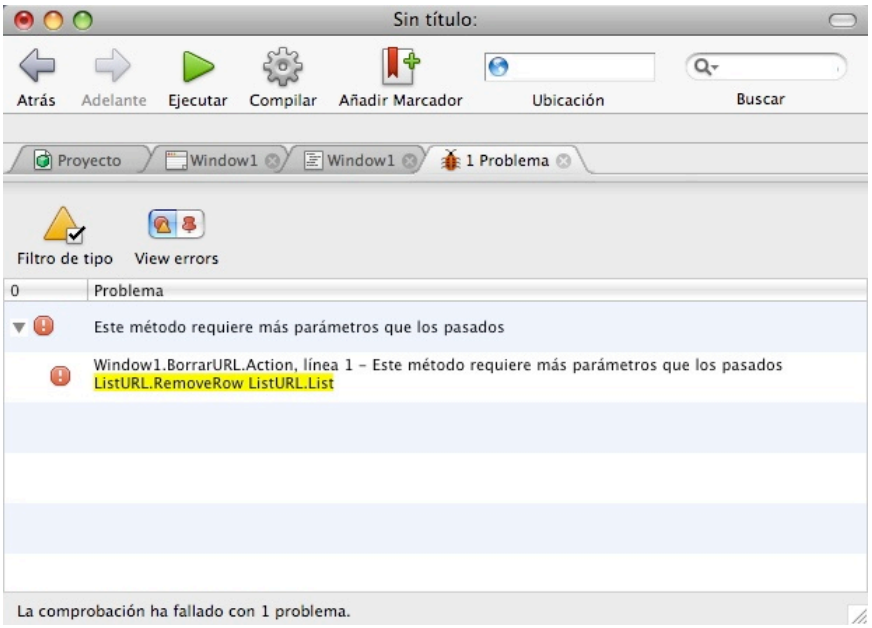
Si la característica Analizar Proyecto encontrase algún problema, abriría un panel “Problemas” y listaría los problemas hallados. A continuación se muestra un ejemplo de un error que evita la compilación del proyecto.

Figura 25. Un error descubierto por el comando Analizar Proyecto.



En el caso de que se encontrasen varios problemas estarían organizados por el tipo de problema en cuestión. Haz clic sobre el triángulo de acceso situado a la izquierda del mensaje de error para ver los errores concretos.

Figura 26. Detalles del error mostrado en la Figura 25.

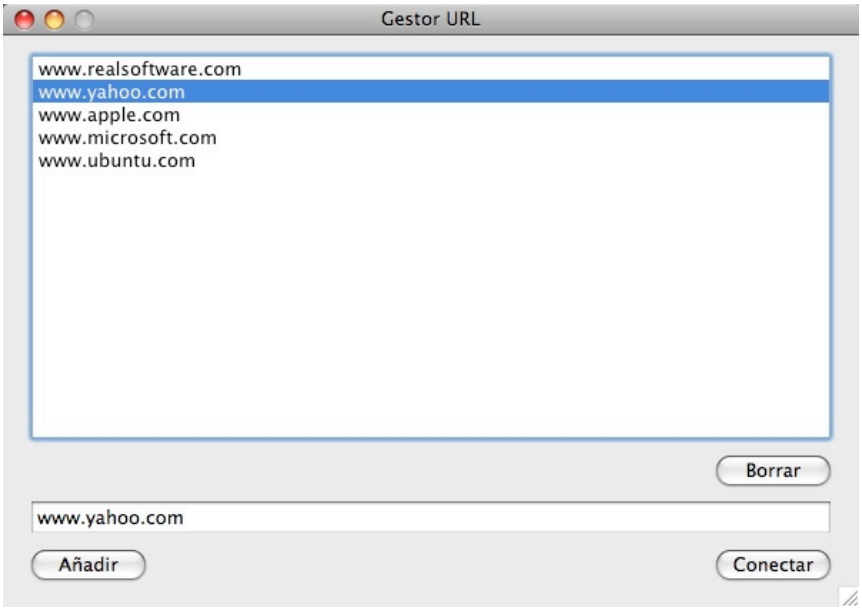


2 (En caso necesario) haz doble clic sobre cada mensaje de error y corrígelo antes de continuar.

3 Haz clic sobre el botón Ejecutar en la Barra de Herramientas Principal para probar la aplicación.

La aplicación mostrará el mismo aspecto que ya hemos visto anteriormente, ¡pero ahora funcionarán todos los controles! Por ejemplo, introduce **http://www.realsoftware.com** en el EditField y haz clic en el botón Añadir. Añade unas cuantos URL al ListBox de este modo y selecciona a continuación cualquiera de ellos en el ListBox para que su contenido pase al EditField.

Figura 27. Al hacer clic sobre un URL éste queda reflejado en el EditField.



Puedes:

- Introducir un URL en el EditField y conectar con el sitio utilizando tu navegador web por omisión.
- Añadir el URL al ListBox.
- Seleccionar cualquier URL en el ListBox.
- Borrar el URL seleccionado en el ListBox.

Si quieres enviar un email, introdúcelo del siguiente modo:

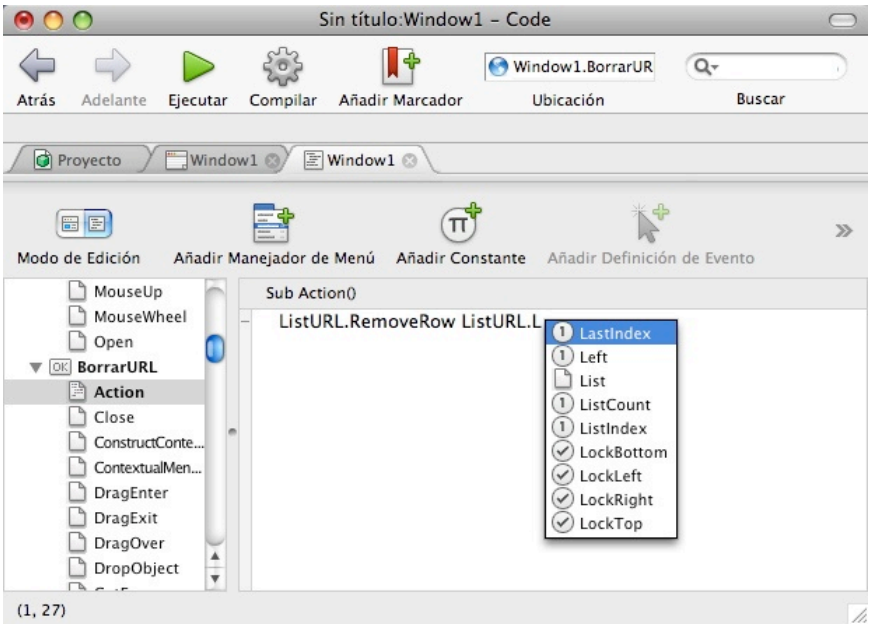
```
mailto: nombreusuario@dominio.com
```

Uso del Autocompletado

Un modo de evitar el uso de términos incorrectos es aprovechando la característica de "autocompletado" de REALbasic. A medida que escribas, REALbasic intentará averiguar lo que estás escribiendo. Si escribes los primeros caracteres de un objeto del lenguaje REALbasic, ya sea interno o una variable, método, o propiedad creada por ti, mostrará su suposición con un tipo de color gris claro. Si la suposición es correcta, pulsa la tecla Tabulador para completar dicha entrada.

En el caso de que exista más de una posibilidad, al pulsar el Tabulador REALbasic mostrará un menú contextual. He aquí un ejemplo:

Figura 28. La característica de autocompletado en acción.



El usuario ha escrito “ListadoURL.L” y pulsado la tecla Tabulador. REALbasic muestra un menú contextual con todos los términos posibles que pueden completar dicha expresión. Utiliza las teclas de cursor Arriba y Abajo para navegar por el listado, y pulsa la tecla Retorno para elegir el elemento seleccionado.

Los pequeños iconos situados a la izquierda de cada término te indican el tipo de objeto de que se trata. En este caso, las marcas de verificación indican que los elementos son propiedades booleanas y los 1 indican que se tratan de propiedades enteras.

¿Cómo se puede mejorar la aplicación?

Aunque los controles hacen su trabajo, intentan hacerlo cuando resulta inapropiado. Por ejemplo,

- Si el EditField está vacío, al hacer clic sobre el botón Añadir se incluirá una línea en blanco sobre el ListBox, y al hacer clic sobre el botón Conectar intentará abrir un URL que no existe.
- Si no está ningún elemento seleccionado en el ListBox, el botón Borrar intentará borrar una fila que no existe.

Deberíamos impedir que se pudiesen pulsar los botones salvo en los casos en los que se cumpla una condición para hacerlo o, como mínimo, dar al usuario algún tipo de explicación.

1. Selecciona Archivo > Salir en Windows o Linux (o MiAplicación.debug > Salir en Mac OS X) para volver al IDE de REALbasic y selecciona la pestaña del editor correspondiente a Window.

2. En el área de navegación del Editor de Código, abre el evento Action del botón Conectar.

   Su contenido actual es:

```
ShowURL URLSeleccionado.Text
```

   Necesitamos que dicha instrucción sea condicional, de modo que la línea se ejecute sólo en el caso de que haya texto en el EditField. Por ahora, tenemos que asumir que el usuario sabe cómo introducir un URL o dirección de correo válidos.

3. Modifica el código para que pase a ser:

```
If URLSeleccionado.Text <> “” then
        ShowURL URLSeleccionado.Text
Else
        MsgBox “Por favor, introduzca un URL o dirección de correo electrónico.”
End if
```

   No hay ningún espacio entre las dobles comillas de la primera línea o entre los signos de menor que y mayor que.

   Este código comprueba en primer lugar si el contenido del EditField está vacío. El símbolo “<>” significa “no es igual a” y las dos comillas sin ningún contenido entre ellas especifica un texto vacío. Esta instrucción if (si) comprueba si el EditField no está vacío. Si el EditField contiene texto, entonces el método ShowURL se ejecutará con normalidad.

   La instrucción Else (si no) se encarga de gestionar el caso donde el EditField está vacío. Si el EditField está vacío, utilizamos el coman-

do MsgBox para mostrar un cuadro de alerta con un mensaje informativo.

4 Expande el elemento BorrarURL en el Editor de Código y selecciona el evento Action.

Necesitamos comprobar la misma lógica. Este código debería funcionar sólo en el caso de que el usuario haya seleccionado un elemento en el ListBox, y no siempre.

5 Modifica el código para que pase a ser como sigue:

```
If ListadoURL.Text <> “” then
        ListadoURL.RemoveRow ListadoURL.ListIndex
Else
        MsgBox “Por favor, seleccione un elemento en el listado.”
End if
```

6 Expande el elemento AnadirURL en el Editor de Código y selecciona el evento Acción.

En este caso, añadiremos una comprobación adicional. Si el elemento ya está seleccionado, entonces no es necesario que se añada de nuevo, de modo que podemos comprobar también dicha condición. La primera comprobación evita que el botón Añadir inserte una fila en blanco y la segunda comprobación evita que se añada una fila duplicada.

7 Modifica el código para que sea como sigue:

```
If URLSeleccionado.Text <> ListadoURL.Text then
        ListadoURL.Addrow URLSeleccionado.Text
Else
        MsgBox “Por favor, introduzca un URL o dirección de correo distinto.”
End if
```

A continuación queremos evitar que el usuario pueda usar los botones Añadir o Conectar en el caso de que el EditField esté vacío.

8 Expande el elemento URLSeleccionado en el Editor de Código y selecciona el manejador correspondiente al evento TextChange en el área del Navegador correspondiente al Editor de Código.

El evento TextChange se ejecuta cada vez que se modifique el texto del EditField. Queremos realizar una acción basada en el estado del EditField justo después de que se haya modificado su texto. En el caso de que el EditField esté vacío, desactivaremos estos dos botones y el usuario no podrá utilizarlos. Si por el contrario hay texto en el EditField, activaremos los botones y haremos que el botón Conectar sea el botón por defecto. El botón por defecto tiene un contorno en torno a él en Windows y Linux y unos "latidos" en Mac OS X.

9 Introduce el siguiente código en el evento TextChange.

```
If Me.Text <> "" then
        ConectarURL.Enabled = True
        ConectarURL.Default = True
        AnadirURL.Enabled = True
Else
        ConectarURL.Enabled = False
        ConectarURL.Default = False
        AnadirURL.Enabled = False
End if
```

Algo que puedes advertir en este código es que la primera línea utiliza el pronombre "Me" (yo) en vez del nombre del control. "Me" siempre se refiere al control al que pertenece el código; dado que está dentro del EditField, URLSeleccionado, "Me" se refiere a URLSeleccionado. En otras palabras, esta línea es equivalente a If URLSeleccionado.Text...

El último paso consiste en modificar el código para el ListBox. Este código necesita comprobar si el usuario ha seleccionado un ele-

mento antes de que intente copiar el texto en el EditField. También debería gestionar los botones Borrar y Conectar. No hay ningún motivo por el que el usuario pueda hacer clic sobre Borrar si no hay ningún elemento seleccionado.

10 Expande el elemento ListadoURL en el Editor de Código y selecciona el evento Change.

11 Modifica el código para que sea como sigue:

```
If Me.Text <> “” then

        URLSeleccionado.Text = ListadoURL.Text

        URLConectado.Enabled = True

        BorrarURL.Enabled = True

Else

        BorrarURL.Enabled = False

End if
```

12 Selecciona Proyecto > Analizar Proyecto para comprobar si has cometido algún error de sintaxis.

Corrige cualquier error y, cuando todo esté bien, procede a probar la aplicación.

13 Haz clic sobre el botón Ejecutar en la Barra de Herramientas Principal para probar la aplicación.

Cuando abras la aplicación puedes probar los mensajes de alerta insertados en cada botón. A continuación añade un URL y comprueba cómo se comporta la aplicación.

14 Cuando hayas terminado, sal de la aplicación de prueba y vuelve al IDE de REALbasic.

Si la Aplicación no Funciona

Esta sección presenta la instrucción If...Else...End if. El Editor de Código se encarga de indentar automáticamente el código contenido por la instrucción If para que resulte más sencillo comprobar el código. Asegúrate de que el código esté indentado tal y como se

muestra en las instrucciones. Asegúrate de que existe una instrucción “End if” por cada “If” y que no escribes “Endif” en vez de “End if” o “end if”. El uso de mayúsculas o minúsculas no es importan, pero el espacio entre “End” e “if” sí. De igual modo, comprueba que cada línea que empiece con una instrucción “if” finalice con un “then”. E igualmente, no deberías dejar un espacio entre las comillas dobles o entre los signos de menor que y mayor que.

Compilando una aplicación independiente

Ahora que ya has finalizado la versión de la aplicación, estás listo para crear una aplicación independiente. La versión *independiente* de la aplicación funciona justo como la aplicación que has estado probando, excepto que no precisará de la aplicación REALbasic propiamente dicha. Podrá ejecutarse haciendo doble clic sobre su icono, tal y como ocurre con cualquier aplicación comercial.

La versión Professional de REALbasic puede crear aplicaciones independientes para las plataformas Windows, Macintosh y Linux. La versión Personal de REALbasic compila aplicaciones independientes sólo para la plataforma sobre la que se esté ejecutando el IDE de REALbasic. También permite compilar versiones demo para todas las plataformas sobre las que *no* se está ejecutando REALbasic. Una versión demo es totalmente funcional pero sale automáticamente una vez hayan transcurrido cinco minutos desde su ejecución.

Compilando tu Aplicación

Compilar una versión independiente de tu aplicación no podría ser más sencillo de lo que lo es en REALbasic.

1 Haz clic en el botón Compilar en la Barra de Herramientas Principal.

Cuando creas una aplicación independiente, REALbasic crea una nueva carpeta con el nombre “Compilaciones” y crea subcarpetas para cada plataforma para la que decidas crear una compilación. Incluirá la aplicación independiente en la subcarpeta correspondiente. La abrirá y seleccionará la aplicación.

Figura 29. El icono de aplicación correspondiente a la compilación de Gestor URL (Windows y Macintosh).

Por omisión, utiliza el nombre "Mi Aplicación" y utiliza el icono de aplicación genérico de cada plataforma. Es posible personalizar de forma sencilla tanto el nombre como el icono.

2 Haz doble clic sobre el icono de "Mi Aplicación" y prueba el programa. Cuando hayas terminado, selecciona Salir para abandonar Gestor URL y vuelve al proyecto de REALbasic.

Personalizar la Aplicación Independiente

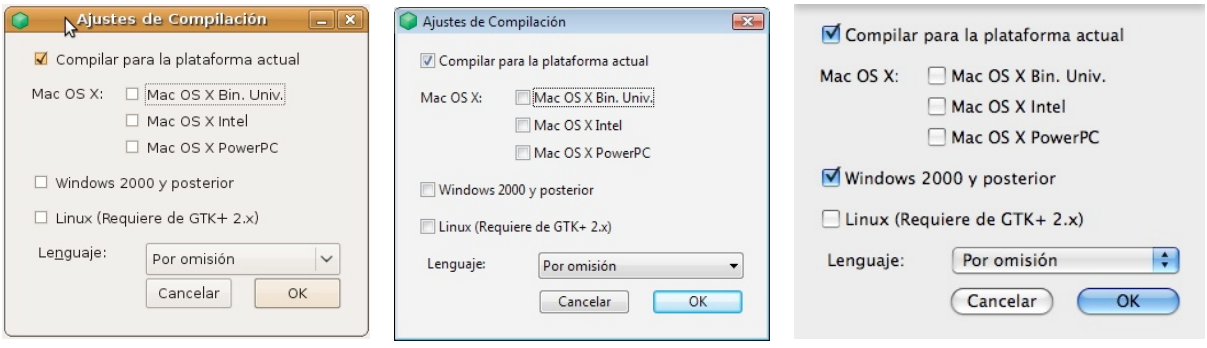
Antes de compilar una aplicación independiente, puedes ajustar varias opciones. Por ejemplo, puedes cambiar el nombre de la aplicación y compilar para otras plataformas.

Puedes seleccionar las plataformas para las que deseas compilar tu aplicación en el cuadro de diálogo Ajustes de Compilación.

1 Selecciona Proyecto > Ajustes de Compilación.

Aparecerá el cuadro de diálogo correspondiente a Ajustes de Compilación, con tu plataforma seleccionada.

Figura 30. Cuadros de diálogo de Ajustes de Compilación (Linux, Vista y Mac OS X).



El cuadro de diálogo de Ajustes de Compilación muestra en la línea inferior la plataforma que se esté utilizando actualmente. Para Macintosh, la selección por omisión, Binario Universal, compila una

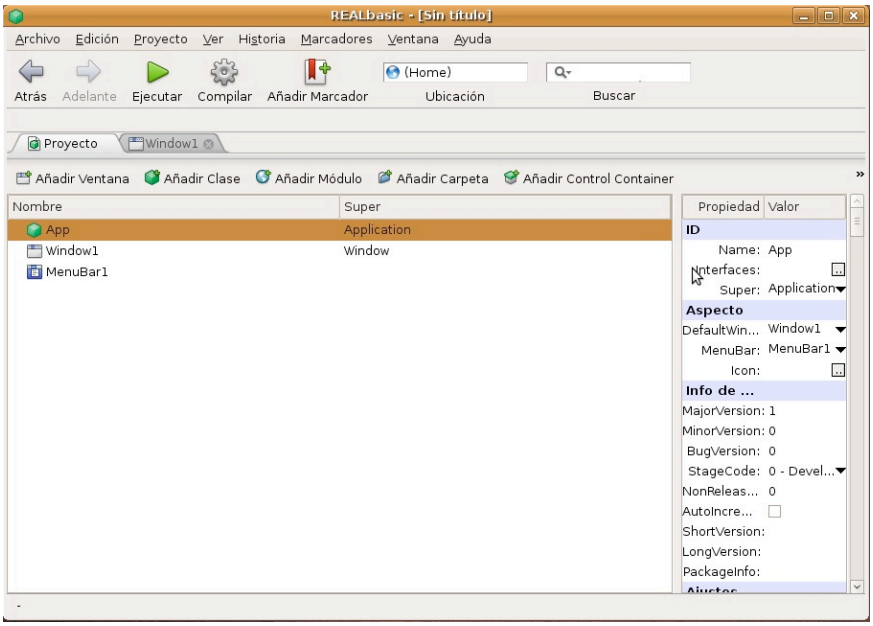
aplicación independiente que funciona de forma “nativa” tanto en los Macintosh basados en PowerPC como Intel.

2 Si dispones de otras plataformas de ordenador, selecciónalas y pulsa sobre OK.

También puedes ajustar las propiedades de la aplicación independiente, incluyendo su nombre. Puedes hacerlo configurando algunas propiedades del objeto *App*. Puedes acceder a este objeto desde el *Editor de Proyecto*.

3 Haz clic en la pestaña Proyecto de la barra Pestañas.

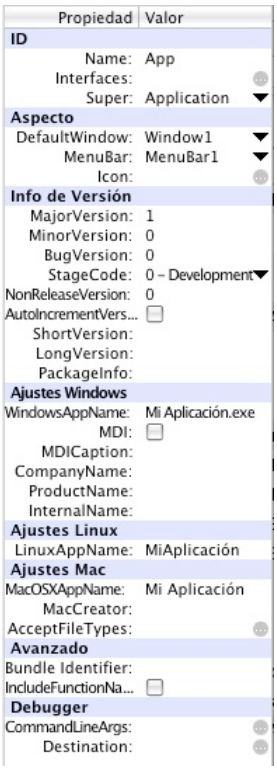
Figura 31. El Editor de Proyecto para la aplicación Gestor URL.



El Editor de Proyecto contiene un listado con los principales componentes de tu aplicación REALbasic. Por ejemplo, incluye los elementos para todas las ventanas utilizadas por tu aplicación, las barra de menús, y los objetos como los sonidos, imágenes, bases de datos, y películas. Por omisión, incluye un elemento para la ventana principal de la aplicación, *Window1*, su barra de menús por omisión, *MenuBar1*, y un elemento asociado con la aplicación como un todo, *App*. Puedes hacer doble clic sobre un elemento en Editor de Proyecto para editarlo o examinarlo.

Cuando seleccionas el objeto App, tal y como se muestra en la Figura 31, el panel Propiedades te muestra las propiedades correspondientes al objeto App.

Figura 32. Las propiedades de la clase App.



En las áreas de ajustes Windows, Linux y Mac, defines el nombre para la aplicación compilada y, en la mayoría de los casos, otra serie de propiedades.

2 Como nombre de la aplicación para el sistema que estés utilizando, introduce Gestor URL.exe para Windows o Gestor URL para Macintosh y Linux.

3 Haz clic sobre el botón Compilar en la Barra de Herramientas Principal.

REALbasic compila la aplicación, guarda la aplicación compilada a un archivo en tu ordenador y abre la ventana que contiene la aplicación compilada.

REALbasic crea una carpeta con el nombre “Compilaciones” y que contiene todas las aplicaciones compiladas. Dentro de esta carpeta Compilaciones se encuentran una serie de carpetas independientes para cada una de las plataformas seleccionadas en el cuadro de diálogo correspondiente a Ajustes de Compilación.

Sal del IDE de REALbasic (Archivo > Salir en Windows y Linux o REALbasic > Salir en Mac OS X).

4. Haz doble clic en la nueva aplicación y pruébala.

5. Si has compilado versiones para otras plataformas, arrástralas sobre dichas plataformas y pruébalas también.

A continuación

El *Tutorial* muestra lo sencillo que resulta desarrollar una aplicación sencilla. Pero hay mucho más dentro de REALbasic. Tal y como se ha podido entrever en el Editor de Proyecto, REALbasic incluye un Editor de Menú y puede soportar múltiples ventanas de diferentes tipos.

REALbasic es un completo sistema de desarrollo orientado a objetos. Con él puedes crear clases (objetos reutilizables) que encapsulen funcionalidad personalizada. Puedes añadirlos al proyecto de modo que puedan reutilizarse en varios lugares del proyecto y guardarlos a disco. En la *Guía del Usuario* aprenderás en detalle el sistema de desarrollo orientado a objetos. La *Referencia del Lenguaje* es un listado explicado de todos los elementos de programación del lenguaje REALbasic.

De igual modo, asegúrate de examinar y probar los proyectos de ejemplo que se suministran con REALbasic, así como la propia sede web de REALbasic en http://www.realsoftware.com, en la que podrás encontrar otros tutoriales y prácticos.